Porting with HPE MPT

Among the Message Passing Interface (MPI) libraries installed on Pleiades, we recommend using HPE's Message Passing Toolkit (MPT) library. Use the command **`module avail mpi-hpe`** to determine which MPT versions are available. You can always load the default MPT version by running **`module load mpi-hpe/mpt`**.

Be sure to load the module in both of the following places:

- On the front-end node when you build your application.
- In the PBS script you use to run jobs on the compute nodes.

Network Considerations

NAS systems use an InfiniBand (IB) network for inter-process remote direct memory access (RDMA) communications. There are two InfiniBand fabrics, designated as ib0 and ib1. In order to maximize performance, we recommend using the ib0 fabric for all MPI traffic. The ib1 fabric is reserved for storage-related traffic. The default configuration for MPT on NAS systems is to use only the ib0 fabric, although the MPI_IB_RAILS environment variable is set to 1+ to allow failover to the ib1 fabric if there are errors using ib0.

TIP: Loading **`mpi-hpe/mpt`** instead of an explicit MPT modulefile allows you to always get the NAS-recommended MPT version and environmental variable settings that enable improvements in InfiniBand stability and scalability.

Environment Variables

When you load an MPT module, several paths (such as CPATH and LD_LIBRARY_PATH) and MPT-related environment variables are reset or modified to non-default values. You can also set other environment variables that may be useful for some applications or for debugging purposes.

Modified Environment Variables

The variables that are modified when you load a module might vary by module version. To find out what environmental variables are set by the modulefile, use the **`module show`** command as follows:

```
% module show mpt_modulefile_name
```

where *mpt_modulefile_name* refers to an explicit MPT modulefile that **`mpi-hpe/mpt`** points to. Some of the variables that are likely to be set to a non-HPE-default value include:

```
MPI_IB_TIMEOUT
MPI_IB_RAILS
MPI_IB_FAILOVER
MPI_IB_RECV_MSGS
MPI_IB_RNR_TIMER
MPI_SYSLOG_COPY
MPI_UD_RECV_MSGS
MPI_UD_TIMEOUT
MPI_WATCHDOG_TIMER
MPIO_LUSTRE_GCYC_MIN_ITER
```

The descriptions of these variables and their HPE default values are as follows:

MPI_IB_TIMEOUT
When an IB card sends a packet, it waits some amount of time for an acknowledgement (ACK) packet to be returned by the receiving IB card. If it does not receive one, it sends the packet again. This variable controls the wait period. The time spent is equal to 4 * 2 ^ MPI_IB_TIMEOUT microseconds.
(HPE) Default: 18

MPI_IB_RAILS
If the MPI library uses the IB driver as the inter-host interconnect, it will by default use a single IB fabric. If this variable is set to 2, the library will try to make use of multiple available separate IB fabrics (ib0 and ib1) and split its traffic across them. If the fabrics do not have unique subnet IDs, then the rail-config utility is required to have been run by the system administrator to enable the library to correctly use the separate fabrics. If this variable is set to 1+, then MPT will set up IB connection on two rails but only send traffic on one; the second rail will be reserved for failover in case of errors.
(HPE) Default: 1

MPI_IB_FAILOVER
When the MPI library uses IB and a connection error is detected, the library will handle the error and restart the connection a number of times equal to the value of this variable. Once there are no more failover attempts left and a connection error occurs, the application will be aborted.
(HPE) Default: 0

MPI_IB_RECV_MSGS
When the MPI library uses IB it must allocate some number of buffers to receive data headers and short messages into. If a rank has all its buffers filled before it can service them, it may go into an error mode. If this happens, try doubling the value of this variable.
(HPE) Default: 512

MPI_IB_RNR_TIMER
When a packet arrives at an InfiniBand HCA and there are no remaining receive buffers for it, the receiving HCA sends a NAK to the requestor. The requesting HCA retries again after some period of time. This variable controls the delay time. Setting a higher value may slow performance in some circumstances, but it will also likely significantly help fabric health during high congestion. See Table 45 of the official InfiniBand specification for precise translations of this value to delay times.
(HPE) Default: 14

MPI_SYSLOG_COPY
When this variable is set, messages about MPT internal errors and system failures are copied to the system log on the machines where they happened. If this variable is set to "2" then MPT internal warnings will also be copied to the syslog.
(HPE) Default: Not set

MPI_UD_RECV_MSGS
This controls the number of InfiniBand UD receive buffers.
(HPE) Default: 800

MPI_UD_TIMEOUT
The IB UD code uses this variable as a timeout to control when to resend UD packets if it has not received some form of ACK from the receiver. The variable is in units of milliseconds.
(HPE) Default: 20

MPI_WATCHDOG_TIMER
If an MPT process has been having continuous IB fabric problems and has not been able to contact another process for this many minutes, then it will abort the application. By default MPT keeps trying to make transfers even in the face of flaky fabrics. This variable may help abort the job sooner if a node has crashed. Set to "0" to disable.
(HPE) Default: 10

MPIO_LUSTRE_GCYC_MIN_ITER

Enables group-cyclic aggregator assignment. Group-cyclic allows multiple aggregators to write to each OST while minimizing lock contention. This value specifies the minimum number of stripes that each aggregator will write so as to ensure the added lock contention of more aggregators is offset by the added bandwidth of more data written per two-phase iteration. A value of "1" will allow assignment of the maximum possible number of aggregators.
(HPE) Default: 0 (not enabled)

See **man MPI** for more information.

Additional Environment Variables

For more MPT-related variables, read **man MPI** after you load an MPT module. Some variables may be useful for certain applications or for debugging purposes. Here are a few of them for you to consider:

MPI_DSM_DISTRIBUTE
: Activates NUMA job placement mode. This mode ensures that each MPI process gets a unique CPU and physical memory on the node with which that CPU is associated. This feature can also be overridden by using **`dplace`** or **`omplace`**. This feature is most useful if running on a dedicated system or running within a cpuset.
Default: Enabled

MPI_MEMMAP_OFF
: Turns off the memory mapping feature, which provides support for single-copy transfers and MPI-2 one-sided communication on Linux. These features are supported for single-host MPI jobs and MPI jobs that span partitions. At job startup, MPI uses the **`xpmem`** module to map memory from one MPI process onto another. The mapped areas include the static region, private heap, and stack. Memory mapping is enabled by default on Linux. To disable it, set the MPI_MEMMAP_OFF environment variable.
Default: Not enabled

MPI_BUFS_PER_PROC
: Determines the number of private message buffers (16 KB each) that MPI is to allocate for each process. These buffers are used to send and receive long messages. Each process has this many buffers for sending with and this many for receiving with. The memory consumed on a shepherd is #ranks_in_the_shepherd * MPI_BUS_PER_PROC * 2 * 16k.
Default: 128 buffers (1 page = 16KB)

MPI_COREDUMP
: Controls which ranks of an MPI job can dump core on receipt of a core-dumping signal. Valid values are NONE, FIRST, ALL, or INHIBIT.
NONE means that no rank should dump core.
FIRST means that the first rank on each host to receive a core-dumping signal should dump core.
ALL means that all ranks should dump core if they receive a core-dumping signal.
INHIBIT disables MPI signal-handler registration for core-dumping signals.
Default: FIRST

MPI_STATS (toggle)
: Enables printing of MPI internal statistics. Each MPI process prints statistics about the amount of data sent with MPI calls during the MPI_Finalize process.
Default: Not enabled

MPI_DISPLAY_SETTINGS
: If set, MPT will display the default and current settings of the environment variables controlling it.
Default: Not enabled

MPI_VERBOSE

Setting this variable causes MPT to display information such as what interconnect devices are being used and what environment variables have been set by the user to non-default values. Setting this variable is equivalent to passing **mpirun** the **-v** option.
Default: Not enabled

MPI_IB_XRC
: When MPT is creating IB connections lazily and using IB host channel adapters (HCAs) that support extended reliable connection (XRC) queue pairs (QPs), MPT will use XRC QPs instead of regular RC QPs. XRC QPs are a more scalable version of the RC QPs currently found on ConnectX IB HCAs.
Default: Enabled

MPI_IB_NUM_QPS
: When InfiniBand RC QPs are being allocated lazily, this limits the number of QPs that any one rank may use. This helps to prevent the host from completely running out of QPs.
Default: (Maximum number of QPs the HCAs can support - 2k) / # of local ranks

MPI_IB_QTIME
: MPT has the ability to use the controlled delay (CoDel) traffic throttling mechanism with Reliable Connection (RC) and User Datagram (UD) InfiniBand protocols. Cluster nodes have the ability to saturate InfiniBand fabrics to the state that they become backed up, start dropping packets, and break all sorts of services like TCP/IP control channels and Lustre. If enabled, MPT will monitor the approximate time that outgoing transfers are queued up at the HCA. If queue time exceeds the value of this variable, then MPT will take measures to significantly reduce the amount of traffic that it is putting onto the fabric for a short period of time. In many cases, this will eventually reduce fabric loading to a reasonable level. This variable is in units of microseconds (us); for example, a value of "200" will cause throttling to begin if minimum queue time exceeds 200us. Suggested values to try are in the range of 100us - 5000us.
Default: Disabled

Building Applications

To build MPI applications with the HPE MPT library, link with **-lmpi** and/or **-lmpi++**. See HPE MPT for some examples.

Running Applications

MPI executable files built with HPE MPT are not allowed to run on the Pleiades front-end (PFE) nodes.

You can run your MPI job on the compute nodes in an interactive PBS session or through a PBS batch job. After loading an MPT module, use the **mpiexec** command to start your MPI processes (do not use **mpirun**).

For example:

```
#PBS -lselect=2:ncpus=20:mpiprocs=20:model=ivy
....
module load mpi-hpe/mpt
mpiexec -np N ./your_executable
```

The **-np** flag (with *N* MPI processes) can be omitted if the value of *N* is the same as the product of the value specified for **select** and the value specified for **mpiprocs**.

Performance Considerations

If your MPI job uses all of the cores in each node (16 MPI processes/node for Sandy Bridge, 20 MPI processes/node for Ivy Bridge, 24 MPI processes/node for Haswell, 28 processes/node for Broadwell, 40 processes/node for Skylake and Cascade Lake, and 128 processes/node for Rome), then pinning MPI processes greatly helps the performance of the code.

By default, the environment variable MPI_DSM_DISTRIBUTE is set to 1 (or true), which will pin processes consecutively on the cores.

If your MPI job does *not* use all the cores in each node, we recommend that you disable MPI_DSM_DISTRIBUTE as follows:

```
setenv MPI_DSM_DISTRIBUTE 0
```

Then, let the Linux kernel decide where to place your MPI processes. If you want to pin processes explicitly, you can try using one of the methods described in Process/Thread Pinning Overview.

The following recommended process pinning method uses the mbind.x tool to pin an eight-MPI-process job with every four processes on four processor cores of a node, using two nodes:

```
#PBS -lselect=2:ncpus=4:mpiprocs=4:model=xxx
mpiexec -np 8 mbind.x ./your_executable
```

Article ID: 100
Last updated: 29 Nov, 2021
Revision: 88
Porting/Building Code -> Porting to NAS Systems -> Porting with HPE MPT
https://www.nas.nasa.gov/hecc/support/kb/entry/100/